Relatório Prática III

Arthur João Lourenço (20100516)
José Daniel Alves do Prado (20103689)
Victor do Valle Cunha (20104135)

Questão 1. Abra o arquivo mandril.bmp no editor hexadecimal em https://hexed.it/ e, analisando o formato do cabeçalho BMP apresentado na Seção 2, indique no relatório: qual é o valor dos campos offset e tamanho do arquivo? Quais são os valores dos componentes de cor R, G e B do primeiro pixel armazenado no arquivo?

Valor do campo offset: 54
Tamanho do arquivo: 786486 bytes
Blue (B) = 12
Green (G) = 11
Red (R) = 09

Questão 2. Qual é o tamanho do cabeçalho do arquivo mandril1.cuif para seu grupo?

26 bytes

Questão 3. No arquivo praticaIII.py tem um função PSNR incompleta. Implemente esta função de maneira a calcular o PSRN passando como parâmetro a imagem original e uma decodificada. Implemente o cálculo do MAE e PSNR com base nas fórmulas da seção 5.

```python
def PSNR(original,decodificada,b):
    try:
        mse = MSE(original,decodificada)
        psnr = 10*math.log10(((2**b-1)**2)/mse)
        return psnr
    except ZeroDivisionError:
        return "Infinito"


tabnine: test | explain | document | ask
def MSE(ori, dec):
    nsymbols = ori.width * ori.height * 3
    r, g, b = (0, 0, 0)
    for i in range(ori.width):
        for j in range(ori.height):
            ori_r, ori_g, ori_b = ori.getpixel((i, j))
            dec_r, dec_g, dec_b = dec.getpixel((i, j))

            r += (ori_r-dec_r)**2
            g += (ori_g-dec_g)**2
            b += (ori_b-dec_b)**2

    return (r+g+b)/nsymbols
```

Questão 4. Indique o PSNR comparando a imagem original mandril.bmp (original) com a imagem obtida a partir do arquivo CUIF.1 (mandril1.bmp). Justifique a resposta do PSNR.

Infinito. Não há perda de dados, pois os bytes são apenas reposicionados de BGRBGRBGR para RRRGGGBBB. Como a soma é comutativa, a diferença dos bytes calculada no loop na função MSE resulta em 0 quando somada na linha 27. Assim, o valor retornado pela função é 0 e, ao tentar dividir para obter o valor de PSNR na linha 10, o resultado é infinito.

Questão 5. Compacte as imagens mandril.bmp e mandril1.cuif com zip. Qual a taxa de compressão obtida para os dois arquivos? Qual arquivo compactou mais? Explique o porquê deste resultado, ou seja, indique a vantagem de organizar os pixels nesta sequência definida pelo CUIF.1 (primeiro os valores de R, depois de G e finalmente de B) para a compressão baseada em RLE ou DPCM? Dica: relembre os princípios da compressão RLE e DPCM e compare a parte de dados de imagem do arquivo mandril.bmp e mandril1.cuif no editor hexadecimal.

Taxa de compressão
Mandril.bmp: 1,044526993x
Mandril1.cuif: 1,090452038x

O arquivo Mandril1.cuif comprimiu mais.

RLE: Como os pixels adjacentes de mesma cor tendem a ser semelhantes ou até mesmo iguais, principalmente em imagens geradas por computador, o formato CUIF tem a vantagem de agrupar esses pixels um ao lado do outro no arquivo, então a repetição de valores é maior, podendo assim ser codificada. Por exemplo, há 6 pixels adjacentes da cor vermelha, idênticos, então na hora de codificar ao invés de apresentar XXXXXX no arquivo, ele pode ser codificado como !6X.

DPCM: Os valores adjacentes são comparados e apenas os erros de predição são quantizados e codificados. Portanto, na extensão CUIF, onde os pixels R, G ou B estão adjacentes, esse algoritmo oferece uma melhor compressão em comparação com o formato BMP, tendo em vista que os pixels adjacentes de mesma cor tendem a ser semelhantes e variar muito pouco entre si. Por exemplo, se tivermos 3 pixels adjacentes, a diferença entre os valores de R tende a ser menor do que a diferença entre os valores de R e G ou R e B, então se os pixels R são 10, 11 e 14, a codificação seria 10, +1, +3.

Questão 6. Agora altere o código em PraticaIII.py para que seja gerado o arquivo mandril2.cuif, que utiliza a versão CUIF.2 (usar 2 em vez de 1 para indicação da versão) e mandril2.bmp. Visualize as imagens mandril.bmp e mandril2.bmp para ver se existem diferenças visíveis. Analise o código que gera o arquivo CUIF.2 (em Cuif.py) e explique o princípio da compressão adotada no CUIF.2.

Notamos que a mandril2.bmp ficou com cores mais fortes (ou mais escuro) do que a mandril.bmp.

O algoritmo funciona compactando os canais de cores G (verde) e B (azul) em um único canal para reduzir a dimensão espacial dos dados da imagem (já sendo um fator de compactação, pois assim ele usa 2 bytes ao invés de 3 bytes). Isso ocorre porque o olho humano é mais sensível à diferença na cor vermelha do que na diferença entre azul e verde, devido à proporção desigual de cones na retina, que medem a frequência de luz: aproximadamente 40:20:1 (R:G:B). Ao combinar os canais G e B, o algoritmo pode aproveitar melhor a menor sensibilidade do olho humano às diferenças entre azul e verde, resultando em uma estratégia de compactação mais eficiente.

Questão 7. Indique as taxas de compressão obtidas pelos CUIF.1 e CUIF.2 para a imagem mandril.bmp? Para este cálculo determine a razão entre um arquivo cuif e a imagem mandril.bmp. Qual versão do CUIF compactou mais?

CUIF.1: 786486/786458 = 1,000035603x

CUIF.2: 786486/524314 = 1,500028609x

A versão CUIF.2 teve uma maior taxa de compressão, pois a razão entre o tamanho original e o tamanho compactado foi maior em comparação com a versão CUIF1.

Questão 8. Indique o PSNR comparando a imagem original mandril.bmp (original) com a imagem obtida a partir do arquivo CUIF.2 (mandril1.bmp). Justifique a resposta do PSNR.

O cálculo do PSNR resultou em 31.00506386112474, indicando que houve uma mudança na imagem e perda de informação, uma vez que esse resultado não é infinito. Isso se dá devido a redução dos bytes por pixel de 3 para 2 ao compactar os canais G e B, mantendo apenas os 4 bits mais significativos de cada um deles, tendo em vista o deslocamento e máscara aplicados.